CS 1337 Course Syllabus

COURSE INFORMATION:

Course Number: CS 1337.007
Course Title: Computer Science I
Course Term: Spring 2026
Schedule: Tu/Th 1:00 – 2:15 PM
Location: ECSS 2.306

INSTRUCTOR CONTACT INFORMATION:

Instructor: Dr. Miguel Razo
Email Address: mrazo@utdallas.edu
Office Location: ECSS 3.605
Office Hours: Tuesday/Thursday: 11 AM - 12 PM in person

GRADER CONTACT INFORMATION:

TA:
TA email:
Office hours:
Location:

Email: When you send an email to me or the grader, please **specify 1337.007 in the subject of the email.**

COURSE PREREQUISITES:

Prerequisite: CS 1436 with a grade of C or better or equivalent.

COURSE DESCRIPTION:

Review of control structures and data types with emphasis on structured data types. Applies the object-oriented programming paradigm, focusing on the definition and use of classes along with the fundamentals of object-oriented design. Includes basic analysis of algorithms, searching and sorting techniques, and an introduction to software engineering. The programming language of choice is C/C++. Students will also be registered for an exam section.

CLASS PARTICIPATION:

Regular class participation is expected regardless of course modality. Students who fail to participate in class regularly are inviting scholastic difficulty. **Regular attendance is highly recommended**. Three consecutive absences or more than five absences lead to a letter grade drop. Seven absences lead to an F. However, **two attendances will be excused/dropped** when computing the final grade to cover the common issues like car-trouble, oversleep, mild sickness. However, if you provide a doctor's medical certificate your absence will be excused. No other reasons will be accepted for an absence excuse.

CLASS MATERIALS:

The instructor may provide class materials including the syllabus, slides, and sample programs will be shared though **UTD box folders** registered for this class as they are intended to supplement the classroom experience. These materials may be downloaded during the course; however, these materials are for registered students' use only. Classroom materials may not be reproduced or shared with those not in class or uploaded to other online environments except to implement an approved Office of Student Accessibility accommodation. Failure to comply with these University requirements is a violation of the Student Code of Conduct.

STUDENT LEARNING OBJECTIVES/OUTCOMES:

After successful completion of this course, the student should have an:

1. Ability to implement simple searching and sorting algorithms.
2. Ability to implement pointers and perform simple memory management.
3. Ability to implement structured data types.
4. Ability to define and implement a class.
5. Ability to use fundamentals of object-oriented design.

Notes: This course is the middle one in the UTDCS programming sequence. Goal is to build your coding expertise in these courses so that you can apply your skills to complete assignments/projects in all the future courses, without much handholding.
CS 1436 Programming Fundamentals (C++)
CS 1337 Computer Science I (C++)
CS 2336 Computer Science II (Java)

Course Learning Outcome	Homework	Exams
1. Ability to use single and multi-dimension arrays	HW1	1
2. Ability to implement simple searching and sorting algorithms.	HW2	1
3. Ability to implement pointers and perform simple memory management	HW3	1
4. Ability to implement structured data types.	HW4	2
5. Ability to define and implement a class.	HW5	3
6. Ability to use fundamentals of object-oriented design.	HW6, Project	3

REQUIRED TEXTBOOKS AND MATERIALS:

Required Textbook:
Starting out with C++. From control structures through objects, Ninth Edition, by Tony Gaddis, Pearson Education, Inc. ISBN: 978-0-13-449837-9.

Codio:
Codio subscription is mandatory.

- 1 Semester - Codio 1-Semester Plan - ISBN: 978-1-7331872-5-1
- Or 2 Semesters - Codio 2 Semester Plan ISBN 978-1-73331872-8-2
- Or Annual - Codio Annual Plan ISBN # 978-1-7331872-1-3

In class activities, homework assignments and practice problems will be posted in Codio, and you will write and submit the solutions there. It is an auto grader tool.

To access codio:

1. Click any Codio assignment link in Blackboard (**Do not** go to the Codio website and create a new account)
2. Subscribe

Software Options:

- Windows:
  - Visual Studio: Community version is OK - https://visualstudio.microsoft.com/downloads/
  - Code::Blocks IDE from codeblocks.org/downloads/binaries - codeblocks-20.03mingw-setup.exe (mingw is important – it is the compiler!) - very easy to install!
- Mac: use pre-installed XCode
- Web based: Online GDB compiler onlinegdb.com/online_c++_compiler (includes debugging!)
- There are several other popular ones out there: incredibuild.com/blog/best-c-ides has a good list!

Additional optional resources:
C++ language tutorial www.cplusplus.com/files/tutorial.pdf
C++ reference: www.cppreference.com
C++ tutorial www.learncpp.com
C++ Gaddis Slides https://utd.link/CS1337

The instructor will use Visual Studio 2022

Important Dates and Times	• First day of class: Tuesday, January 20st, 2026 • Exam 1: Thursday, February 26th, 2026 10% weightage • Exam 2: Friday, March 13th, 2026 25% weightage • Exam 3: Friday, May 8th, 2026 25% weightage

Tentative Course Calendar

Week	Dates (M W)	Class Activity/Notes	ICs Assigned	HWs Assigned	Chapters
1	Jan 20th, 22th	Review of syllabus/1436	Quiz	HW0	Syllabus
2	Jan 27th, 29th	Review of 1436 Files/arrays	ICA 1	HW1	Ch 1-7
3	Feb 3rd, 5th	Search and Sorting	ICA 2	HW2	Ch 8
4	Feb 10th,12th	Pointers	ICA 3		Ch 9
5	Feb 17th, 19th	C Strings	ICA 4		Ch 10
6	Feb 24th, **26th**	C Strings, Review	ICA 5	HW3	**Exam 1: on eLearning during class time**
7	March 3rd, 5th	Structured Data		HW4	Ch 11
8	March 10th, 12th, **March 13th**	Introduction to classes, Mid semester Review	ICA 6		**Exam 2: Friday 2 hours 4-6 PM** 25% weightage
9	**March 17th, 19th**	**Spring Break**			
10	March 24th, 26th	Classes, Constructors,	ICA 7		Ch13
11	March 31st, April 2nd	UML, Copy constructor, Operator Overloading	ICA 8	HW5	Ch 14
12	April 7th, 9th	Inheritance	ICA 9		Ch 15
13	April 14th, 16th	Polymorphism, Pure Virtual Functions	ICA 10		Ch 15
14	April 21st, 23rd	Exceptions, Recursion, Project	ICA 11	HW6	Ch 16
15	April 28th, 30th	Project, Linked Lists	ICA 12		**Project Due!**
16	May 5th, 7th **May 8th**	Review			**Exam 3: Friday 2 hours 4-6 PM** 25% weightage

*The descriptions and timelines contained in this syllabus are **subject** to change at the discretion of the instructor.*

Letter grades will be assigned as follows:

[97,100]	A+	[94,97)	A	[90,94)	A-
[87,90)	B+	[84,87)	B	[80,84)	B-
[77,80)	C+	[74,77)	C	[70,74)	C-
[67,70)	D+	[64,67)	D	[60,64)	D-
Below 60	F				

Grading Criteria	In class participation/assignments: 10% Exam 1: 10%, Exam 2: 25%, Exam 3: 25%, Homework Assignments: 20%, A Small Project: 5%, Attendance 5% **To enroll in course CS/CE 2336, you must have completed CE/CS 1337 with a grade of C or higher). So, you need to score an average of 70% (C-) across both exams 2 and 3 to qualify for a C or better in the course, e.g. if your final score is say B- but your exams score is D, you final score will still be a C-. You can pass the course but may not be eligible to move to 2336.** Historically students who skip programming assignments, or do not put much effort into their programming assignments, or get a lot of help from classmates, mentors, or others, do not perform well on exam questions testing the material covered by the assignment
In Class and Homework Assignments	In **Class Assignments (ICAs)** are given every week. **Weekly Assignments will be due the following Wednesday at midnight.** They will have equal weightage – all of them together will contribute 10% to the final weighted grade. **Homework Assignments (HWs)** are announced during class, according to the academic calendar. They also have the same weightage and contribute 20% to the final grade. The complexity level of each assignment will vary – each assignment may take several hours to complete. You are expected to start working on them as soon as they are posted so that you have "enough" time to work through the glitches, get the necessary help & still manage to submit on time. **Late submissions are not allowed.** If you cannot complete an assignment due to medical condition, send the Doctor note to the professor. You will be given a few additional days to complete the assignment. Each week's assignment may vary from multiple small programs to one large program OR somewhere in-between. You are expected to spend several hours at a computer every week. The right way to approach the programming assignments is to start on them right away & get help when you get stuck (you can approach the instructor, TA, or tutors at CS mentor center for help). Do not waste lots of hours trying to fix one specific issue. In simple words, your approach will determine whether programming assignments provide an enjoyable learning experience or end up like a painful activity that ruins your self-confidence. You can develop & test your program using any C++ IDE, but you need to pass all the official testcases in Codio to complete. Assignments are auto graded through Codio & the scores will appear automatically in eLearning. Your program should be as generic as possible – it should be able to handle all possible valid input values and output meaningful results. As time permits, TA will manually review your code in Codio and give feedback. All submissions are subject to random manual inspection as well - you should NOT use any concepts that are not yet covered in the course yet. You should NOT write code just to pass the specified testcases either. Your assignment score may be reduced to 0 for such violations. We are all here to learn! Sophisticated tools are available in Codio and beyond to detect plagiarism. Suspicious cases will be referred to UTD administration directly - Review http://utdallas.edu/conduct/integrity & http://utdallas.edu/conduct/manage-dishonesty for details.

Diagnostic Test	1. **Mandatory Diagnostic Tests**: All students enrolled in the 1337 course must take the diagnostic tests. 2. **Purpose of Test Results**: The results will help students understand their competency levels. This approach will ensure all students reach the same level of competency. Based on their scores, they will be required to attend sessions in CSMC. 3. This test will be part of your first assignment. You will need to attend the **remedial Sessions in CSMC based on your test score**: • **Above 90%**: Students don't need to attend any session, and they automatically receive 100% in their assignment. • **80-90%**: Students must attend one session in CSMC to receive 100%. • **70-80%**: Students must attend two sessions in CSMC to receive 100%. • **60-70%**: Students must attend three sessions in CSMC to receive 100%. • **Below 50%**: Students must attend all sessions in CSMC to receive 100%. 4. CSMC will give us an update on the sessions attended and your assignment score will be updated accordingly 5. **Remedial Sessions Timing:** CSMC will conduct the remedial sessions during the 2nd and 3rd weeks of the semester. All students are required to attend these sessions based on the above criteria.
Two Day Rule	Students are welcome to ask questions about homework assignments, projects, or exams via email, in person, or through MS Teams **starting from the day the task is posted or announced**. This continues **until two days before the deadline**. During the **final two days before the deadline**, the instructor will only be available to answer questions **during scheduled office hours**. Please note: This rule applies **regardless of whether the last two days fall on weekdays or weekends**.
Make-up Exams	Make-up examinations will be administered **only for well-documented emergencies**. A student must make every attempt possible, via email, to notify the instructor that he/she will miss a scheduled exam **prior** to the scheduled date and time or **immediately** thereafter. **If notification is not received in a timely manner, no make-up will be given.**
Project	There will be one final project. You will implement the project using Object Oriented Design concepts. More details will be provided in the middle of the semester.
Late Work	**Late submissions will not be** accepted
Computer:	You will need a computer for this class. Minimum specs can be found here: https://oit.utdallas.edu/technologyinitiative/hardware. However, if you do not have a computer, the CS computer lab is open 24/7 and contains all the software you will need for this class. Use of smart phones, headphones/earphones, cameras or microphones is strictly prohibited. Tablets that use a digital pen (no keyboards or mice) are allowed for students to take notes. Laptops are allowed only during the second part of the class. There is no need to copy my examples in class as I will make them available to you. I would rather you focus on how we are solving the problem
Class Attendance	Regular attendance is highly recommended. **Three consecutive absences or five absences lead to a letter grade drop. Eight absences lead to an F.** **If you are sick, your absence will be excused only if you give a doctor's note.** This will not be counted against your 5 absences for a letter grade drop. **If a student attends all classes, he/she will get two bonus points over the final grade.**

Classroom Citizenship	The instructor encourages students to take active part in class discussions. No question is too simple/stupid to be asked. So, do not hesitate. **Use of smart phones and headphones/earphones is strictly prohibited. Laptops need to be closed**, until I request to open for an in-class activity.
UT Dallas Syllabus Policies and Procedures	The information contained in the following link constitutes the University's policies and procedures segment of the course syllabus. Please go to http://go.utdallas.edu/syllabus-policies for these policies.

While working on your programming assignments:

Step	Task	Comments
1	Pseudocode	High level explanation of steps at the top of program/each function
2	Test Cases	Define a set of inputs and expected outputs
3	Source Code Design	Overall design of the program
4	Run Test Cases	Ensure your program produces the expected output
5	Execution	Use Codio to t
6	Polish your Source Code	Formatting and coding style, assign meaningful names for variables and functions

What you need to do to be successful in this course:

Ask for help at any time. If you do not understand something or are having trouble implementing a concept. The sooner you ask that question, the sooner you will get an answer. That answer will allow you to move forward. I want you to succeed; don't be afraid to ask questions.

- The instructor is available to help during office hours.
- Be proactive.
  - Don't wait till the day an assignment is due to seek help. Please note that I do not have office hours every day. If you wait till close to the assignment due date to seek my help, it is possible that I won't have any office hours that day or there may be many students who will be competing for my time when you come to my office hours. **You may not email your code to the instructor or grader expecting us to find your errors.**
  - You can also ask help at CSMC https://csmc.utdallas.edu/
  - Don't wait till the end of the semester to seek help. If you have gotten far behind in your coursework or have done significant damage to your course average, I may not be able to help.

COMPUTER SCIENCE MENTORING CENTER (CSMC)

The Computer Science Mentoring Center (CSMC) is a free resource available to all students taking this class.

The CSMC provides assistance in many areas including:

- Understanding core concepts related to this class
- Developing a logical framework for a program
- Connecting programming constructs to the logic of the program
- Assisting in solving syntax and logical errors in your code
- Exam reviews and reworks (by faculty request)

The mentors will meet with you 1-on-1 to address your specific problem areas. Their goal is to help you understand what is wrong and how to fix it, but they will not do the work for you. For more information about

the CSMC, including location and hours of operation, please visit http://csmc.utdallas.edu

Take responsibility for your education.

- Read your assigned reading before the lecture before course meetings. You are expected to have an understanding of the assigned textbook material before meetings
- Attend every meeting and pay close attention.
- Dedicate 9-10 hours per week outside of class meetings to CS 1337 for reading, watching recordings, practicing writing code, assignments, exercises and studying for exams.
- I will give challenging assignments to push you toward learning general concepts, developing critical thinking and core programming skills. Part of being a professional is learning how to teach yourself. I am going to guide you through the topics of the semester, but a significant portion of what you take with you to the next class will be things that you have learned on your own.
- The more programs you practice with outside of lecture the better you will do in this course. I will also show you samples of programs and of the use of programming constructs / patterns. I will introduce you to program development methodologies. However, you learn to program by doing – coding, testing, and fixing (debugging).
- Enter the sample programs from the text. Experiment by making small changes. Note how the changes affect the program translation and/or execution.
  The more programs you practice with outside of lecture the better you will do in this course. I can teach you the syntax of the C++ programming language and about typical programming constructs. I will also show you samples of programs and of the use of programming constructs / patterns. I will introduce you to program development methodologies. However, you learn to program by doing – coding, testing, and fixing (debugging).
- Complete the Checkpoint questions at the end of the sections of the text.
- Complete the Review Questions and Exercises at the end of the chapters of the text.
- Pick a few of the Programming Exercises at the end of the chapter and write programs that satisfy the requirements given. This is good practice for the types of coding/MC questions I will ask you on the exam.
  Start your assignment immediately. All assignments are designed to be worked on over a period of days. I expect that you will work on the assignment a little at a time rather than waiting until a day or two before it is due. Those that procrastinate will find this class to be much harder than it should be and will face the risk of below average grades.

Practice time management skills. Good time management is necessary for this class.

- Start your assignment immediately. All assignments are designed to be worked on over a period of days. I expect that you will work on the assignment a little at a time rather than waiting until a day or two before it is due. Those that procrastinate will find this class to be much harder than it should be and will face the risk of below average grades.
- I expect everyone to devote at least an hour a day to this class. Doing this will help you to divide tasks up into chunks and work a little at a time on an assignment rather than waiting until a day or two before it is due. You will have a very difficult time succeeding in this class if you schedule to finish every assignment at the last minute.

Attend every class. You are paying for an education. Don't waste your money by skipping class. I will give you everything you need to complete projects and do well on the tests. You have to be there to get the information.

Make mistakes! This is how you learn. Don't be discouraged when something goes wrong. Programming takes lots of practice and mistakes will always happen. Study the mistakes you made so that you can learn from them for the future.

What each student should expect in this course:

A problem solving class. This class is not a programming class. Computer science is all about problem solving. The content of this class is to teach you how to solve problems using a computer. In order to solve those problems, you will need to learn a foreign language (C++) and write solutions that the computer can interpret.

An open environment dedicated to learning. I want students to feel free to voice their opinions. Oftentimes as we code in class, I will ask students what they would do in a certain situation. I want each student to feel as if he/she can speak freely and also be open for other students to discuss that idea, even if that means that some students will disagree.
Exams focused on application. Many of the questions I ask on an exam require you to apply your knowledge to answer the question. This may involve finding errors in code or determining output of a code segment. I expect you to apply the knowledge you have learned to the situations on the test. Questions on the test are designed to make sure that you understand what you are doing rather than repeating an example from your notes or the textbook.
A simulated professional experience. The projects in this class require you to exercise strategies found in "the real world". Your logic for a project may force you to learn new techniques that haven't yet been discussed in class. You will have to perform code maintenance and improve the efficiency of previously written code. These things offer a small taste of how life might be once you graduate and are given large sums of money by a company seeking your skills.
A deep understanding of C++ and object-oriented programming. My goal is for you to know all the topics of CS 1337. You should have peace of mind moving on in your program because you will be fully prepared to tackle what the next course in the sequence will throw at you.

Academic Integrity:

All assignments, exercises and exams are to be individual efforts. You are not to collaborate with other students. Prior to the assignment due date, you are not to: discuss assignment solutions with other students, distribute your code to others, or publish your code. Copying of programming assignments, exercises, or exams, in whole or in part, from other students will be considered an act of scholastic dishonesty. Copying assignments from previous semesters will be considered an act of scholastic dishonesty.

For programming assignments, you may use source code provided by the instructor. You are not to view, copy, or distribute code from any other sources, including code from other students, code from assignments submitted in past semesters, or code from the Internet. Plagiarism detection software will be employed to detect copying of code.

Grading Concerns:

If you think there is a mistake in the grading of your assignment or exercise and would like to request that it be regraded, **you must notify both the grader and the instructor (email the grader and copy the instructor)** of this by email within **one week** after the date the grade is posted in the grade book on eLearning. Your request for any regrade must describe in detail what you perceive as the problem with the grading. Keep in mind that a regrade may result in an increase or in a reduction of the original grade.

Most deductions are made because students did not fully read the assignment instructions, did not adequately test their programs, or did not follow the style guidelines provided. You may not change the problem to suit your purposes. Most assignments restrict the use of programming constructs and library functions not covered in lecture, others require that you use constructs or functions. To get the maximum credit you must read the directions carefully and test your programs thoroughly.

COMET CREED:

This creed was voted on by the UT Dallas student body in 2014. It is a standard that Comets choose to live by and encourage others to do the same:

"As a Comet, I pledge honesty, integrity, and service in all that I do."

UT DALLAS SYLLABUS POLICIES AND PROCEDURES:

The information contained in the following link constitutes the University's policies and procedures segment of the course syllabus.

Please go to http://go.utdallas.edu/syllabus-policies for these policies.

The descriptions and timelines contained in this syllabus are subject to change at the discretion of the Professor.